# Deep Learning – Quiz 1

**Q1. You are given a data set on cancer detection. You've build a classification model and achieved an accuracy of 96%. Why shouldn't you be happy with your model performance? What can you do about it?**

**Ans:-** If you have worked on enough data sets, you should deduce that cancer detection results in imbalanced data. In an imbalanced data set, accuracy should not be used as a measure of performance because 96% (as given) might only be predicting majority class correctly, but our class of interest is minority class (4%) which is the people who actually got diagnosed with cancer. Hence, in order to evaluate model performance, we should use Sensitivity (True Positive Rate), Specificity (True Negative Rate), F measure to determine class wise performance of the classifier. If the minority class performance is found to to be poor, we can undertake the following steps:

1. We can use undersampling/oversampling to balance data.
2. We can use a metric other than accuracy (something like precision/recall, which will give a better picture of the model's utility)
3. We can alter the prediction threshold value by doing probability calibration and finding an optimal threshold using AUC-ROC curve.
4. We can assign weight to classes such that the minority classes gets larger weight.
5. We can also use anomaly detection.

**Q2. You came to know that your model is suffering from low bias and high variance. What approach should you use to tackle it? Why?**

**Ans:-** Low bias occurs when the model's predicted values are near to actual values. In other words, the model becomes flexible enough to mimic the training data distribution. While it sounds like great achievement, but not to forget, a flexible model has no generalization capabilities. It means, when this model is tested on an unseen data, it gives disappointing results. In such situations, we can use bagging algorithm (like random forest) to tackle high variance problem. Bagging algorithms divides a data set into subsets made with repeated randomized sampling. Then, these samples are used to generate a set of models using a single learning algorithm. Later, the model predictions are combined using voting (classification) or averaging (regression).

Also, to combat high variance, we can:

1. Use regularization technique, where higher model coefficients get penalized, hence lowering model complexity.
2. Reduce the complexity of the model (drop-connect/drop-out, if neural-net) to avoid over-fitting.
3. Use top-n features from variable importance chart. May be, with all the variable in the data set, the algorithm is having difficulty in finding the meaningful signal.

**Q3. You've built a random forest model with 10000 trees. You got delighted after getting training error as 0.00. But, the validation error is 34.23. What is going on? Haven't you trained your model perfectly?**

Ans:- The model has trained "too perfectly": it has overfitted. Training error 0.00 means the classifier has mimicked the training data patterns to an extent, that they are not available in the unseen data. Hence, when this classifier was run on unseen sample, it couldn't find those patterns and returned prediction with higher error. In random forest, it happens when we use larger number of trees than necessary. Hence, to avoid these situation, we should tune number of trees using cross validation.

**Q4. You are working on a classification problem. For validation purposes, you've randomly sampled the training data set into train and validation. You are confident that your model will work incredibly well on unseen data since your validation accuracy is high. However, you get shocked after getting poor test accuracy. What went wrong?**

Ans:- In case of classification problem, we should always use stratified sampling instead of random sampling. A random sampling doesn't takes into consideration the proportion of target classes. On the contrary, stratified sampling helps to maintain the distribution of target variable in the resultant distributed samples also. Another way to fix this would be to run multiple shuffled-iterations (k-fold cross validation) to make sure it's not a case of a lucky-validation data distribution.

**Q5. What is the difference between Generative Models and Discriminative Models?**

Ans:- A generative model will learn categories of data while a discriminative model will simply learn the distinction between different categories of data. Discriminative models will generally outperform generative models on classification tasks. For example, a variational autoencoder is a generative model, while a CNN is a discriminative model.

**Q6. Derive the cross-entropy loss function for a 3-class classification problem?**

$$-\sum_{i}^{n}( y'_1 log(y_1) + y'_2 log(y_2) + y'_3 log(y_3))$$

$$y'_1 + y'_2 + y'_3 = 1, \qquad y_1 + y_2 + y_3 = 1 \qquad (\textit{sum of probabilities})$$

So,

$$-\sum_{i}^{n}( y'_1 log(y_1) + y'_2 log(y_2) + (1 - y'_1 - y'_2) log(1 - y_1 - y_2))$$

$$-\sum_{i}^{n}( y'_1 log(\tfrac{y_1}{1-y_1-y_2}) + y'_2 log(\tfrac{y_2}{1-y_1-y_2}) + log(1 - y_1 - y_2))$$